

IDO-EVB3562-V2 Android 开发手册

注意事项

1 源码获取

2 Android_SDK编译环境配置

3 SDK编译

3.1 一键编译

3.2 单独编译

3.2.1 uboot编译步骤

3.2.2 kernel编译步骤

3.2.3 Android编译步骤

4 驱动开发

4.1 LOGO旋转

4.2 触摸旋转

4.3 系统旋转



www.industio.cn

IDO-EVB3562-V2 Android 开发手册

深圳触觉智能科技有限公司

www.industio.cn

文档修订历史

版本	PCBA版本	修订内容	修订	审核	日期
V1.0	V2A	创建文档	HJT		2024/12 /30

注意事项

源码的解压和编译使用普通用户即可，无需使用sudo或root用户。

1 源码获取

本文档以Android13为例

链接: <https://pan.baidu.com/s/1nL9GXMxfE85eR9RO1meYw?pwd=1234>

提取码: 1234

从网盘下载SDK源码文件到PC端的Linux主机中，如下图所示：

名称	修改日期	类型	大小
 ID0_EVB3562_V2_Android13.tar.gzaa	2025/3/1 16:51	GZAA 文件	4,194,304 KB
 ID0_EVB3562_V2_Android13.tar.gzab	2025/3/1 16:54	GZAB 文件	4,194,304 KB
 ID0_EVB3562_V2_Android13.tar.gzac	2025/3/1 16:57	GZAC 文件	4,194,304 KB
 ID0_EVB3562_V2_Android13.tar.gzad	2025/3/1 16:59	GZAD 文件	4,194,304 KB
 ID0_EVB3562_V2_Android13.tar.gzae	2025/3/1 17:02	GZAE 文件	4,194,304 KB
 ID0_EVB3562_V2_Android13.tar.gzaf	2025/3/1 17:05	GZAF 文件	3,983,875 KB
 evb3562_v2_sdk.md5	2025/3/2 10:37	MD5 文件	1 KB

校验下载文件的完整性，命令如下：

```

1  $ md5sum -c evb3562_v2_sdk.md5

```

结果如下图所示：

```

$ md5sum -c evb3562_v2_sdk.md5
ID0_EVB3562_V2_Android13.tar.gzaa: 成功
ID0_EVB3562_V2_Android13.tar.gzab: 成功
ID0_EVB3562_V2_Android13.tar.gzac: 成功
ID0_EVB3562_V2_Android13.tar.gzad: 成功
ID0_EVB3562_V2_Android13.tar.gzae: 成功
ID0_EVB3562_V2_Android13.tar.gzaf: 成功
ID0_EVB3562_V2_Android13.tar.gzag: 成功
ID0_EVB3562_V2_Android13.tar.gzah: 成功

```

创建一个文件夹（名字自拟），将下载的文件解压到 RK3562 目录，命令如下：

```

1  $ mkdir RK3562-V2
2  $ cat ID0_EVB3562_V2_Android13.tar.gza* | tar -xz -C RK3562-V2
3  $ cd RK3562-V2/ID0_EVB3562_V2_Android13/

```

解压后的目录下有.git隐藏文件，使用以下命令从.git中恢复SDK源码，命令如下：

```

1  $ git reset --hard

```

结果如下图所示：

```
ID0_EVB3562_V2_Android13$ git reset --hard
正在更新文件: 100% (1047287/1047287), 完成.
HEAD 现在位于 43353fc ID0 EVB3562 V2 SDK TAG V1.0
/ID0_EVB3562_V2_Android13$ ls
Android.bp      bootable        build.sh        development     hardware        libcore         mkimage.sh      prebuilts      rkst            test            vendor
android_t.conf  bootstrap.bash  cts            device         javaenv.sh      libnativehelper packages         restore_patches.sh RKTools        toolchain      WORKSPACE
art            build          dalvik         external        kernel          mkcombinedroot  pdk             rkbin          sdk            tools
bionic         BUILD          developers     frameworks     kernel-5.10     mkimage_ab.sh   platform_testing RKDocs         system        u-boot
```

2 Android_SDK编译环境配置

推荐编译主机配置如下:

1. Ubuntu22.04 操作系统64位
2. 64 位 CPU
3. 16GB 物理内存+交换内存
4. 250GB 空闲的磁盘空间

开发环境搭建, 请参考RKDocs\android\Rockchip_Android13_SDK_Developer_Guide_CN.pdf文档中的附录A编译开发环境搭建, 安装OpenJDK 8和一些编译依赖软件, Ubuntu22.04通用软件安装包, 命令如下:

```
Bash
1 $ sudo apt-get update
2 $ sudo apt-get install openjdk-8-jdk
3 $ sudo apt-get install git gnupg flex bison gperf libssl1.2-dev libbsd-jav
a \
4 squashfs-tools build-essential zip curl libncurses5 libncurses5-dev zlib1g
-dev \
5 pngcrush schedtool libxml2 libxml2-utils xsltproc lzop libc6-dev schedtool
g++-multilib \
6 lib32z1-dev lib32ncurses-dev lib32readline-dev gcc-multilib libswitch-perl
libssl-dev \
7 unzip zip device-tree-compiler liblz4-tool python2 python3-pyelftools -y
```

结果如下图所示:

```

$ sudo apt-get update
命中:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
命中:2 http://mirrors.tuna.tsinghua.edu.cn/ubuntu jammy InRelease
命中:3 http://mirrors.tuna.tsinghua.edu.cn/ubuntu jammy-updates InRelease
命中:4 http://mirrors.tuna.tsinghua.edu.cn/ubuntu jammy-backports InRelease
正在读取软件包列表... 完成
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801$ sudo apt-get install openjdk-8-jdk
正在读取软件包列表... 完成
正在分析软件包的依赖关系树... 完成
正在读取状态信息... 完成
openjdk-8-jdk 已经是最新版 (8u402-ga-2ubuntu1~22.04)。
升级了 0 个软件包，新安装了 0 个软件包，要卸载 0 个软件包，有 41 个软件包未被升级。
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801$ sudo apt-get install git gnupg flex bison gperf libssl-dev libbsd-java \
> squashfs-tools build-essential zip curl libncurses5 libncurses5-dev zlib1g-dev \
> pngcrush schedtool libxml2 libxml2-utils xsltproc lzop libc6-dev schedtool g++-multilib \
> lib32z1-dev lib32ncurses-dev lib32readline-dev gcc-multilib libswitch-perl libssl-dev \
> unzip zip device-tree-compiler liblz4-tool python2 python3-pyelftools -y
正在读取软件包列表... 完成
正在分析软件包的依赖关系树... 完成
正在读取状态信息... 完成
bison 已经是最新版 (2:3.8.2+dfsg-1build1)。
build-essential 已经是最新版 (12.9ubuntu3)。
device-tree-compiler 已经是最新版 (1.6.1-1)。
flex 已经是最新版 (2.6.4-8build2)。
g++-multilib 已经是最新版 (4:11.2.0-1ubuntu1)。
gcc-multilib 已经是最新版 (4:11.2.0-1ubuntu1)。
lib32readline-dev 已经是最新版 (8.1.2-1)。
lzop 已经是最新版 (1.04-2build2)。
python3-pyelftools 已经是最新版 (0.27-1)。
squashfs-tools 已经是最新版 (1:4.5-3build1)。
zip 已经是最新版 (3.0-12build2)。
gperf 已经是最新版 (3.1-1build1)。
libbsd-java 已经是最新版 (0.0.7-5)。
liblz4-tool 已经是最新版 (1.9.3-2build2)。
libssl1.2-dev 已经是最新版 (1.2.15+dfsg2-6)。
libswitch-perl 已经是最新版 (2.17-2.1)。
pngcrush 已经是最新版 (1.8.13-0.1)。
python2 已经是最新版 (2.7.18-3)。
schedtool 已经是最新版 (1.3.0-4)。
curl 已经是最新版 (7.81.0-1ubuntu1.16)。
git 已经是最新版 (1:2.34.1-1ubuntu1.10)。
gnupg 已经是最新版 (2.2.27-3ubuntu2.1)。
lib32ncurses-dev 已经是最新版 (6.3-2ubuntu0.1)。
lib32z1-dev 已经是最新版 (1:1.2.11.dfsg-2ubuntu9.2)。
libc6-dev 已经是最新版 (2.35-0ubuntu3.7)。
libncurses5-dev 已经是最新版 (6.3-2ubuntu0.1)。
libssl-dev 已经是最新版 (3.0.2-0ubuntu1.15)。
libxml2 已经是最新版 (2.9.13+dfsg-1ubuntu0.4)。
libxml2-utils 已经是最新版 (2.9.13+dfsg-1ubuntu0.4)。
unzip 已经是最新版 (6.0-26ubuntu3.2)。
xsltproc 已经是最新版 (1.1.34-4ubuntu0.22.04.1)。
zlib1g-dev 已经是最新版 (1:1.2.11.dfsg-2ubuntu9.2)。
libncurses5 已经是最新版 (6.3-2ubuntu0.1)。
升级了 0 个软件包，新安装了 0 个软件包，要卸载 0 个软件包，有 41 个软件包未被升级。

```

3 SDK编译

3.1 一键编译

进入 sdk 根目录执行命令，可以使用编译脚本，一次编译uboot、kernel、system等分区，命令如下：

```

1 $ cd ID0_EVB3562_V2_Android13/
2 $ source build/envsetup.sh
3 $ lunch rk3562_t-userdebug

```

结果如下图所示：

```

/ID0_EVB3562_V2_Android13$ source build/envsetup.sh
/ID0_EVB3562_V2_Android13$ lunch rk3562_t-userdebug

=====
PLATFORM_VERSION_CODENAME=REL
PLATFORM_VERSION=13
TARGET_PRODUCT=rk3562_t
TARGET_BUILD_VARIANT=userdebug
TARGET_BUILD_TYPE=release
TARGET_ARCH=arm64
TARGET_ARCH_VARIANT=armv8-a
TARGET_CPU_VARIANT=cortex-a53
TARGET_2ND_ARCH=arm
TARGET_2ND_ARCH_VARIANT=armv8-a
TARGET_2ND_CPU_VARIANT=cortex-a53
HOST_ARCH=x86_64
HOST_2ND_ARCH=x86
HOST_OS=linux
HOST_OS_EXTRA=Linux-5.15.0-122-generic-x86_64-Ubuntu-20.04.5-LTS
HOST_CROSS_OS=windows
HOST_CROSS_ARCH=x86
HOST_CROSS_2ND_ARCH=x86_64
HOST_BUILD_TYPE=release
BUILD_ID=TQ3C.230805.001.B2
OUT_DIR=out
=====

```

编译命令如下：

-d 参数指定要编译的dts文件为ido-evb3562-v2a-dsi-mipi

```

▼ Plain Text
1 $ ./build.sh -UCKAu -d ido-evb3562-v2a-dsi-mipi -J6

```

结果如下图所示：

```

ID0_EVB3562_V2_Android13$ ./build.sh -UCKAu -d ido-evb3562-v2a-dsi-mipi -J6
will build u-boot
will build kernel with Clang
will build kernel
will build android
will build update.img
-----KERNEL_VERSION:5.10
-----KERNEL_DTS:ido-evb3562-v2a-dsi-mipi
Force use clang and llvm to build kernel-5.10

```

uboot编译成功，如下图所示：

```

Platform RK3562 is build OK, with new .config(make rk3562_defconfig -j128)
/ID0_EVB3562_V2_Android13/prebuilts/gcc/linux-x86/aarch64/gcc-linaro-6.3.1-2017.05-x86_64_aarch64-linux-gnu/bin/
aarch64-linux-gnu-
2025年 01月 20日 星期一 15:04:24 CST
/ID0_EVB3562_V2_Android13
Build uboot ok!

```

编译kernel成功，如下图所示：

```
#### build completed successfully (03:13 (mm:ss)) ####
Build external wifi driver ok!
有那个文件或目录
package resource.img with charger images
Pack ./tools/images/ & ../kernel-5.10/resource.img to resource.img ...
Unpacking old image(../kernel-5.10/resource.img):
rk-kernel.dtb logo.bmp logo_kernel.bmp 3
Pack to resource.img succeeded!
Packed resources:
rk-kernel.dtb battery_1.bmp battery_2.bmp battery_3.bmp battery_4.bmp battery_5.bmp battery_fail.bmp logo.bmp logo_kernel.bmp battery_0.bmp 10
./resource.img with battery images is ready
```

编译Android成功，如下图所示：

```
/rk3562_t/product.img --sparse --output out/target/product/rk3562_t/super.img"
2025-01-20 18:39:03 - build_super_image.py - INFO : Done writing image out/target/product/rk3562_t/super.img
#### build completed successfully (01:40:27 (hh:mm:ss)) ####
Build android ok!
```

编译完成后结果，如下图所示：

```
----- OK -----
*****rkImageMaker ver 2.23*****
Generating new image, please wait...
Writing head info...
Writing boot file...
Writing firmware...
Generating MD5 data...
MD5 data generated successfully!
New image generated successfully!
Making update.img OK.
Make update image ok!
```

固件所在目录为/rockdev/Image-rk3562_t/，update.img为整包固件，结果如下图所示：

```
/ID0_EVB3562_V2_Android13$ ls rockdev/Image-rk3562_t/
baseparameter.img boot.img dtbo.img misc.img pcba_small_misc.img recovery.img super.img update.img
boot-debug.img config.cfg MiniLoaderAll.bin parameter.txt pcba_whole_misc.img resource.img uboot.img vbmeta.img
```

3.2 单独编译

3.2.1 uboot编译步骤

进入 sdk 根目录执行命令。使用编译脚本编译，命令如下：

Bash

```
1 $ cd ID0_EVB3562_V2_Android13/
2 $ source build/envsetup.sh
3 $ lunch rk3562_t-userdebug
4 $ ./build.sh -U
```

编译成功结果，如下图所示：

```
2025年 01月 20日 星期一 19:16:21 CST
.../ID0_EVB3562_V2_Android13
Build uboot ok!
```

手动执行命令编译，命令如下：

Shell

```
1 $ cd u-boot
2 $ ./make.sh rk3562
```

结果如下图所示：

```
.../ID0_EVB3562_V2_Android13$ cd u-boot/
.../ID0_EVB3562_V2_Android13/u-boot$ ./make.sh rk3562
## make rk3562_defconfig -j128
#
# configuration written to .config
#
scripts/kconfig/conf --silentoldconfig Kconfig
CHK include/config.h
CFG u-boot.cfg
GEN include/autoconf.mk.dep
CFG spl/u-boot.cfg
CFG tpl/u-boot.cfg
GEN include/autoconf.mk
GEN spl/include/autoconf.mk
```

编译结果，如下图所示：

```
Platform RK3562 is build OK, with new .config(make rk3562_defconfig -j128)
.../ID0_EVB3562_V2_Android13/prebuilts/gcc/linux-x86/aarch64/gcc-linaro-6.3.1-2017.05-x86_64_aarch64-linux-gnu/bin/aarch64-linux-gnu-
2025年 01月 20日 星期一 19:55:48 CST
```

3.2.2 kernel编译步骤

内核配置文件路径：kernel-5.10/arch/arm64/configs/
设备树文件路径：kernel-5.10/arch/arm64/boot/dts/rockchip/
内核配置及设备树文件名，列表如下：

序号	文件名	功能说明
----	-----	------

1	ido-evb3562-v2a-lvds.dts	显示为LVDS屏
2	ido-evb3562-v2a-dsi-mipi.dts	显示为MIPI屏

以编译MIPI屏dts为例，编译命令如下：

▼Shell

```
1  ./build.sh -CK -d ido-evb3562-v2a-dsi-mipi -J6
```

```
ID0_EVB3562_V2_Android13$ ./build.sh -CK -d ido-evb3562-v2a-dsi-mipi -J6
will build kernel with Clang
will build kernel
-----KERNEL_VERSION:5.10
-----KERNEL_DTS:ido-evb3562-v2a-dsi-mipi
Force use clang and llvm to build kernel-5.10

=====
PLATFORM_VERSION_CODENAME=REL
PLATFORM_VERSION=13
TARGET_PRODUCT=rk3562_t
TARGET_BUILD_VARIANT=userdebug
TARGET_BUILD_TYPE=release
TARGET_ARCH=arm64
TARGET_ARCH_VARIANT=armv8-a
TARGET_CPU_VARIANT=cortex-a53
TARGET_2ND_ARCH=arm
TARGET_2ND_ARCH_VARIANT=armv8-a
TARGET_2ND_CPU_VARIANT=cortex-a53
HOST_ARCH=x86_64
HOST_2ND_ARCH=x86
```

编译成功结果，如下图所示：

```
Pack ./tools/images/ & ../kernel-5.10/resource.img to resource.img ...
Unpacking old image(../kernel-5.10/resource.img):
rk-kernel.dtb logo.bmp logo_kernel.bmp 3
Pack to resource.img succeeded!

Packed resources:
rk-kernel.dtb battery_1.bmp battery_2.bmp battery_3.bmp battery_4.bmp battery_5.bmp battery_fail.bmp logo.bmp logo_kernel.bmp battery_0.bmp 10
../resource.img with battery images is ready
/home/fuyingzhe/work/RK3562/Android/work/share/ID0_EVB3562_V2_Android13
Repacking header 2 boot...
```

单独编译kernel生成可直接烧录的boot.img

此处的编译方法的前提已存在rockdev/Image-rk3562_t/boot.img文件（即Android代码已经完全编译过一次或者执行过build.sh -K）。

编译的原理：在kernel目录下将编译生成的 kernel.img 和 resource.img 替换到旧的 boot.img 中，命令如下：

▼Shell

```
1  $ cd kernel-5.10
2  $ export PATH=../prebuilts/clang/host/linux-x86/clang-r450784d/bin:$PATH
3  $ alias msk='make CROSS_COMPILE=aarch64-linux-gnu- LLVM=1 LLVM_IAS=1'
4  $ msk ARCH=arm64 rockchip_defconfig android-13.config rk356x.config
```

结果如下图所示：

```
/ID0_EVB3562_V2_Android13$ cd kernel-5.10/
/ID0_EVB3562_V2_Android13/kernel-5.10$ export PATH=../prebuilts/clang/host/linux-x86/clang-r450784d/bin:$PATH
/ID0_EVB3562_V2_Android13/kernel-5.10$ alias msk='make CROSS_COMPILE=aarch64-linux-gnu- LLVM=1 LLVM_IAS=1'
/ID0_EVB3562_V2_Android13/kernel-5.10$ msk ARCH=arm64 rockchip_defconfig android-13.config rk356x.config
#
# configuration written to .config
#
Using .config as base
Merging ./kernel/configs/android-13.config
Value of CONFIG_DEVMEM is redefined by fragment ./kernel/configs/android-13.config:
Previous value: CONFIG_DEVMEM=y
New value: # CONFIG_DEVMEM is not set

Value of CONFIG_NET_ACT_POLICE is redefined by fragment ./kernel/configs/android-13.config:
Previous value: # CONFIG_NET_ACT_POLICE is not set
New value: CONFIG_NET_ACT_POLICE=y

Value of CONFIG_NET_ACT_BPF is redefined by fragment ./kernel/configs/android-13.config:
Previous value: # CONFIG_NET_ACT_BPF is not set
New value: CONFIG_NET_ACT_BPF=y

Value of CONFIG_NET_CLS_MATCHALL is redefined by fragment ./kernel/configs/android-13.config:
Previous value: # CONFIG_NET_CLS_MATCHALL is not set
New value: CONFIG_NET_CLS_MATCHALL=y

Value of CONFIG_NET_SCH_TBF is redefined by fragment ./kernel/configs/android-13.config:
Previous value: # CONFIG_NET_SCH_TBF is not set
New value: CONFIG_NET_SCH_TBF=y

Value of CONFIG_GKI_HIDDEN_GPU_CONFIGS is redefined by fragment ./kernel/configs/android-13.config:
Previous value: # CONFIG_GKI_HIDDEN_GPU_CONFIGS is not set
New value: CONFIG_GKI_HIDDEN_GPU_CONFIGS=y

Value of CONFIG_UNICODE_NORMALIZATION_SELFTEST is redefined by fragment ./kernel/configs/android-13.config:
Previous value: # CONFIG_UNICODE_NORMALIZATION_SELFTEST is not set
New value: CONFIG_UNICODE_NORMALIZATION_SELFTEST=y
#
# merged configuration written to .config (needs make)
#
# configuration written to .config
#
Using .config as base
Merging ./kernel/configs/rk356x.config
Value of CONFIG_MALI_CSF_SUPPORT is redefined by fragment ./kernel/configs/rk356x.config:
```

命令如下：

```
▼ Shell |
1 $ msk ARCH=arm64 BOOT_IMG=../rockdev/Image-rk3562_t/boot.img ido-evb3562-v2
a-dsi-mipi.img -j6
```

结果如下图所示：

```
/ID0_EVB3562_V2_Android13/kernel-5.10$ msk ARCH=arm64 BOOT_IMG=../rockdev/Image-rk3562_t/boot.img ido-evb3562-v2a-dsi-mipi.img -j6
SYNC include/config/auto.conf.cmd
CALL scripts/atomic/check-atomics.sh
CALL scripts/checksyscalls.sh
CHK include/generated/compile.h
CHK kernel/kheaders_data.tar.xz
LD [M] drivers/net/wireless/rockchip_wlan/rkwifi/bcmndrv/bcmndrv.ko
LD [M] drivers/net/wireless/rockchip_wlan/rkwifi/bcmndrv/dhd_static_buf.ko
Image: resource.img (with ido-evb3562-v2a-dsi-mipi.dtb logo.bmp logo.kernel.bmp) is ready
Image: boot.img (../rockdev/Image-rk3562_t/boot.img + Image) is ready
Image: zboot.img (../rockdev/Image-rk3562_t/boot.img + Image.lz4) is ready
### build completed successfully (24 seconds) ###
```

使用此方法编译出kernel-5.10/boot.img文件可以直接用于烧录至boot分区。

3.2.3 Android编译步骤

Android编译命令如下：

```
▼ Shell |
1 $ source build/envsetup.sh
2 $ lunch rk3562_t-userdebug
3 $ make -j6
```

结果如下图所示

```
/ID0_EVB3562_V2_Android13$ source build/envsetup.sh
/ID0_EVB3562_V2_Android13$ lunch rk3562_t-userdebug

=====
PLATFORM_VERSION_CODENAME=REL
PLATFORM_VERSION=13
TARGET_PRODUCT=rk3562_t
TARGET_BUILD_VARIANT=userdebug
TARGET_BUILD_TYPE=release
TARGET_ARCH=arm64
TARGET_ARCH_VARIANT=armv8-a
TARGET_CPU_VARIANT=cortex-a53
TARGET_2ND_ARCH=arm
TARGET_2ND_ARCH_VARIANT=armv8-a
TARGET_2ND_CPU_VARIANT=cortex-a53
HOST_ARCH=x86_64
HOST_2ND_ARCH=x86
HOST_OS=linux
HOST_OS_EXTRA=Linux-5.15.0-122-generic-x86_64-Ubuntu-20.04.5-LTS
HOST_CROSS_OS=windows
HOST_CROSS_ARCH=x86
HOST_CROSS_2ND_ARCH=x86_64
HOST_BUILD_TYPE=release
BUILD_ID=TQ3C.230805.001.B2
OUT_DIR=out
=====
```

编译成功结果，如下图所示：

```
2025-01-20 18:38:59 - build_super_image.py - INFO : Building super image from info dict...
2025-01-20 18:38:59 - common.py - INFO : Running: "/mnt/5c953a98-9ee6-45b9-8cea-a2898eb313d7/tuyz/RK3562/Android/work/share/ID0_EVB3562_V2_Android13/out/hos
t/linux-x86/bin/lpmake --metadata-size 65536 --super-name super --metadata-slots 2 --device super:3263168512 --group rockchip_dynamic_partitions:3258974208 --par
tition system:readonly:1003139072:rockchip_dynamic_partitions --image system=out/target/product/rk3562_t/system.img --partition system_dkkm:readonly:262144:rockc
hip_dynamic_partitions --image system_dkkm=out/target/product/rk3562_t/system_dkkm.img --partition system_ext:readonly:177844224:rockchip_dynamic_partitions --im
age system_ext=out/target/product/rk3562_t/system_ext.img --partition vendor:readonly:281157632:rockchip_dynamic_partitions --image vendor=out/target/product/rk3
562_t/vendor.img --partition vendor_dkkm:readonly:28151808:rockchip_dynamic_partitions --image vendor_dkkm=out/target/product/rk3562_t/vendor_dkkm.img --partitio
n odm:readonly:749568:rockchip_dynamic_partitions --image odm=out/target/product/rk3562_t/odm.img --partition odm_dkkm:readonly:262144:rockchip_dynamic partition
s --image odm_dkkm=out/target/product/rk3562_t/odm_dkkm.img --partition product:readonly:268189696:rockchip_dynamic_partitions --image product=out/target/product
/rk3562_t/product.img --sparse --output out/target/product/rk3562_t/super.img"
2025-01-20 18:39:03 - build_super_image.py - INFO : Done writing image out/target/product/rk3562_t/super.img

### build completed successfully (01:40:27 (hh:mm:ss)) ###
Build android ok!
```

固件所在目录为rockdev/Image-rk3562_t/, 结果如下图所示：

```
/ID0_EVB3562_V2_Android13$ ls rockdev/Image-rk3562_t/
baseparameter.img  boot.img  dtbo.img  misc.img  pcba_small_misc.img  recovery.img  super.img  update.img
boot-debug.img    config.cfg  MiniLoaderAll.bin  parameter.txt  pcba_whole_misc.img  resource.img  uboot.img  vbmeta.img
```

4 驱动开发

4.1 LOGO旋转

修改kernel-5.10/logo.bmp 和kernel-5.10/logo_kernel.bmp，LOGO旋转270度。

4.2 触摸旋转

触摸旋转代码修改如下：

```

1  --- a/kernel-5.10/arch/arm64/boot/dts/rockchip/ido-evb3562-v2a-dsi-mipi.dts
2  +++ b/kernel-5.10/arch/arm64/boot/dts/rockchip/ido-evb3562-v2a-dsi-mipi.dts
3  @@ -360,8 +360,8 @@ gt911@14 {
4      pinctrl-0 = <&touch_gpio>;
5      goodix_irq_gpio = <&gpio0 RK_PB6 IRQ_TYPE_LEVEL_LOW>;
6      goodix_rst_gpio = <&gpio0 RK_PB5 GPIO_ACTIVE_HIGH>;
7  -    // touchscreen-inverted-y;
8  -    // touchscreen-swapped-x-y;
9  +    touchscreen-inverted-y;
10 +    touchscreen-swapped-x-y;
11     status = "okay";
12 };
13 };

```

结果如下图所示：

```

&i2c3 {
    status = "okay";
    gt911@14 {
        compatible = "goodix,gt9xx";
        reg = <0x14>;
        pinctrl-names = "default";
        pinctrl-0 = <&touch_gpio>;
        goodix_irq_gpio = <&gpio0 RK_PB6 IRQ_TYPE_LEVEL_LOW>;
        goodix_rst_gpio = <&gpio0 RK_PB5 GPIO_ACTIVE_HIGH>;
        // touchscreen-inverted-y;
        // touchscreen-swapped-x-y;
        touchscreen-inverted-y;
        touchscreen-swapped-x-y;
        status = "okay";
    };
};

```

4.3 系统旋转

系统旋转代码修改如下：

```

1  --- a/device/rockchip/rk3562/BoardConfig.mk
2  +++ b/device/rockchip/rk3562/BoardConfig.mk
3  @@ -26,7 +26,7 @@ PRODUCT_KERNEL_CONFIG += rk356x.config
4   # used for fstab_generator, sdmmc controller address
5   PRODUCT_BOOT_DEVICE := ff870000.mmc
6
7  -SF_PRIMARY_DISPLAY_ORIENTATION := 0
8  +SF_PRIMARY_DISPLAY_ORIENTATION := 270
9
10 # Disable emulator for "make dist" until there is a 64-bit qemu kernel
11 BUILD_EMULATOR := false
12

```

结果如下图所示：

```

# BOARD_AVB_ENABLE := true
# used for fstab_generator, sdmmc controller address
PRODUCT_BOOT_DEVICE := ff870000.mmc

# SF_PRIMARY_DISPLAY_ORIENTATION := 0
SF_PRIMARY_DISPLAY_ORIENTATION := 270

# Disable emulator for "make dist" until there is a 64-bit qemu kernel
BUILD_EMULATOR := false
TARGET_BOARD_PLATFORM := rk3562
TARGET_BOARD_PLATFORM_GPU := mali-g52
TARGET_RK_GRALLOC_AIDL := true
TARGET_RK_GRALLOC_VERSION := 4
BOARD_USE_DRM := true

```

以上三个位置修改后，重新编译，编译命令如下：

```

1  $ source build/envsetup.sh
2  $ lunch rk3562_t-userdebug
3  $ ./build.sh -CKAu -d ido-evb3562-v2a-dsi-mipi -J6

```